Twenty Years of High Performance, Parallel Computing: Vector Machines to HPC Linux Clusters

Harold Trease
Computation Science and Mathematics Division
Fundamental Sciences Directorate
Pacific Northwest National Laboratory

Overview

- This is “not” meant to be a stroll down memory lane.
- Hopefully, I can convey some useful lessons learned.

- I work at the application software level.
- I try to influence the development of “middleware”.
- I have to live with the delivered hardware.

- I want to do application software development using a “shared memory” programming model, but have access to high-performance parallel computing resources for solving application problems. And I won’t be happy until I can do this at the petabytes/petaflop scale.

Why do this level of HPC computing ?

- Scientific Discovery through Advanced Computing
  - Science drivers
  - Multi-scale (time and/or space) resolution
  - Complex physics

- Computational science reasons:
  - Benchmarking
  - Turnaround time
  - Access of large memory (🡐)
  - Cycle sucking parasite

Overview

- Review of the hardware
- Software of the software
  - Codes
  - Languages
  - Parallel Programming Paradigms
- Environments
- Soapbox Issues
- Applications
- Lessons Learned

Hardware Background
(8 strong oxen .vs. 10,000 chickens)

- LANL:
  - CRAY Series [1s, XMP, YMP] at LANL (PCs days) [80's]
  - Connection Machines [CM2/CM5] [late 80's early 90's]
  - Cray T3D/T3E series [early – middle 90's]
  - SGI ASCI Blue Mountain [middle - late 90's]

- PNNL:
  - IBM SP2 [early 2000's]
  - PC Cluster Computing [early 2000's]
  - ORNL IBM/Power4 and PNNL HP (current)

Hardware Background

- CRAYs: Evolving, stable software development environment, 64-bit data types, gather-scatter, multi-tasking
- CM2: Data Parallel, SIMD, RTS, F90, Global operations
- CM5 [MIMD, MPI]
- T3D/T3E [shmem]
- SGI Blue Mountain [Terabytes/Teraflops, dedicated time for terascale application runs, billion element meshes]

- IBM SP2 [32-bits]
- PC Cluster Computing [Price/Performance]
- HP Super Cluster [TBD, but 10s of Terabytes/Teraflops]

Software Background

- Computational mesh based methods:

  - Unstructured 3-D meshes, compressed sparse matrix data structures (unassembled/assembled), dynamic in space and time.

- Computational geometry

  ➔ Computational meshes

  ➔ Computational physics solvers

- One semi-continuous computational physics code development effort.

  - FLM2D ➔FLM3D➔POLLY➔X3D➔NWGrid➔P3D

  - ~400 man-years, 2-3 million lines of core code

  - 40-50 developers have added/contributed code

Languages

- 80's: F77, CRAY style pointers, dynamic memory allocation, runtime database management support
- 90's: CMFortran, C++(BS), F90, Python
- 2000's: Components using a mixture of languages (F77/F90, C, C++, Python, Java) through the Common Component Architecture (CCA)

Programming Paradigms

- Multiple vector processes that parallel process through time-slicing and communicate through disk files. [CRAY-1]
- Vector/Multitasking [CRAY-XMP/YMP]
- Data Parallel, SIMD [CM2]
- MIMD, MPI [CM5, T3D/E] (the dark times)
- Data Parallel, MIMD [SGI, HP]

(My) Programming Paradigm

- Domain decomposition (partitioning)
  - Load balancing
  - Maximum locality of data
- Data parallel processor-to-processor communication through a library interface called Global Arrays.
- Language
  - F77/F90, C, C++
  - “Embedded” Python with shared memory access
  - Runtime, memory resident relational database (RDBMS)
- Components (CCA)

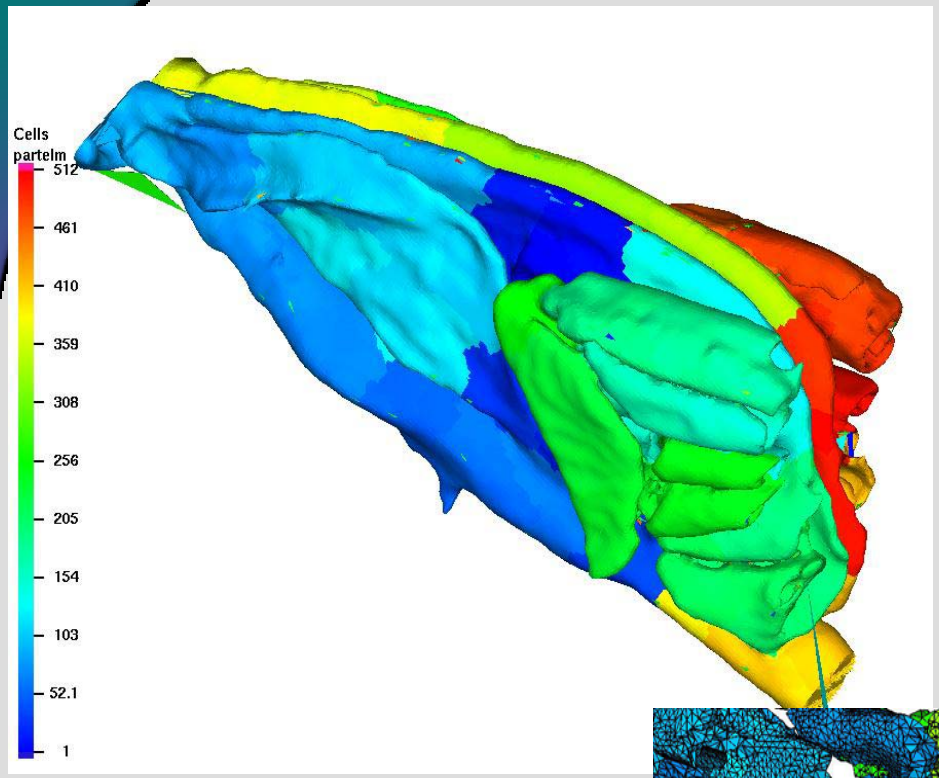
Global Arrays: A data parallel communication library for managing distributed shared memory data structures.

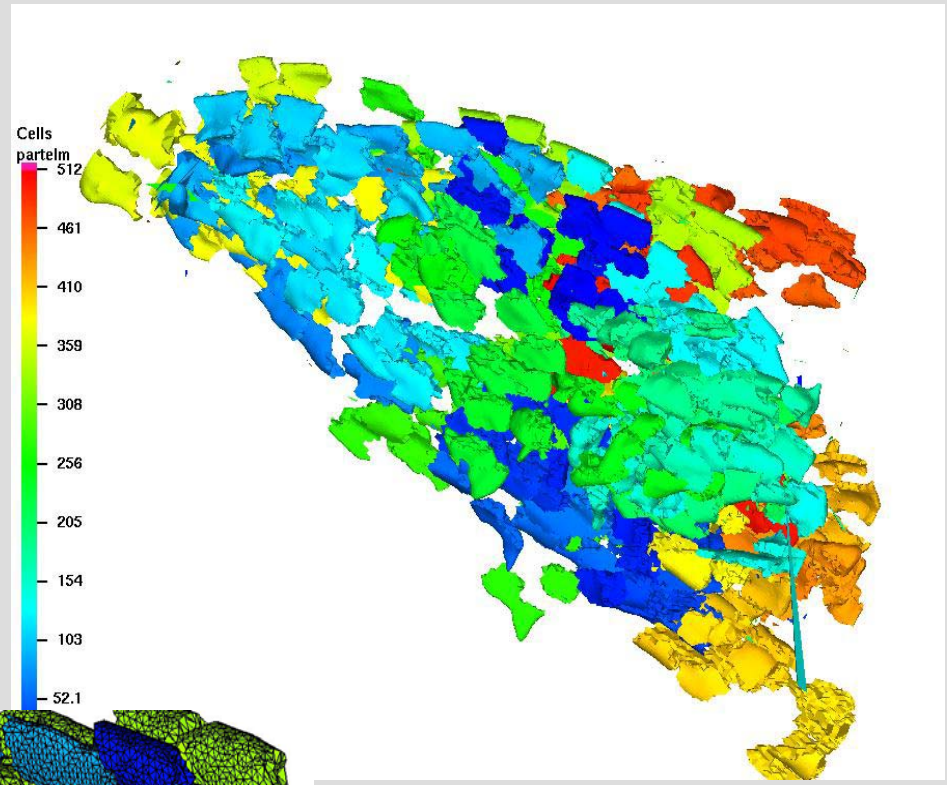
- Gets/Puts, Gather/Scatter, etc.
- Non-uniform data layouts

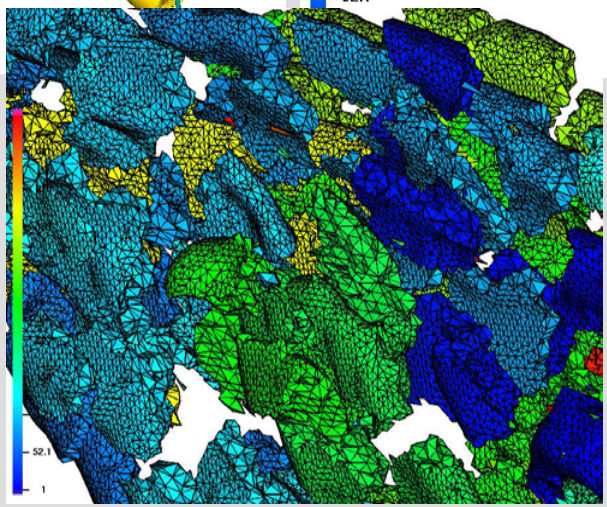
- Scatter_with_sum
- Segmented Scans (operators=sum, copy, max/min)
- Compress/Uncompress
- Two key sorter (e.g., used to map from nodes/elements to rows and columns of a compressed sparse matrix)

- M X N mapping

Maximize Locality of Data using METIS







Environments

- Operating Systems, Schedulers, Batch Management:
  - Important, but I don't have much impact on the choice?
  - Always report security issues, but never report other anomalies because they might come in handy.
- I/O Issues: I prefer global shared disks, but don't require them.
- Compilers: Pretty good for what they do. They only serve the purpose of giving me access to my RDBMS.

Environments (cont)

- Multi-processor, Dynamic Debugging Environment:
  - Why can't we have one available before a machine reaches the end of it's lifetime ??
  - Checkpoint/Restart (available on early CRAYs)

- Distributed Graphics/Visualization: Solutions get tough for terascale datasets (like billion elements meshes and mesh data).

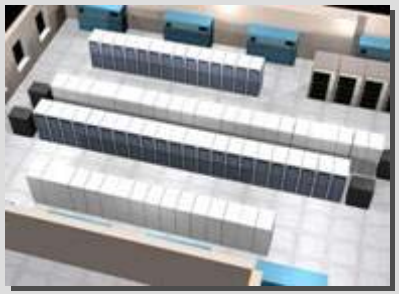
Computational Infrastructure and Applications

- DOE Office of Science
- PNNL EMSL
- HPCS2 and PNNL Computing Infrastructure

The DOE Office of Science has three centers with distinct architectural features to enable scientific discovery through advanced computing

EMSL-HP

11 teraflops



- HP Itanium2
- 1,980 processors
- 6 GFLOP processor
- 98% DGEMM efficiency
- 125 cabinets

NERSC-IBM

10 teraflops

- IBM Power3
- 6,656 processors
- 1.5 GFLOP processor
- 87% DGEMM efficiency
- 104 cabinets

ORNL-Cray/IBM

3.2 teraflops

- Cray X1
- 256 processors
- 12.8 GFLOP processor
- ?? DGEMM efficiency
- 8 cabinets

4.5 teraflops

- IBM Power4
- 864 processors
- 5.2 GFLOP processor
- 65% DGEMM efficiency
- 30 cabinets

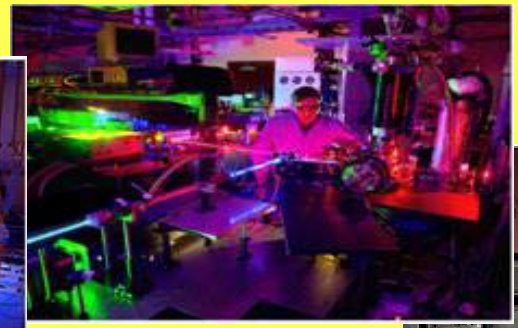
Three Centers - Three Vendors

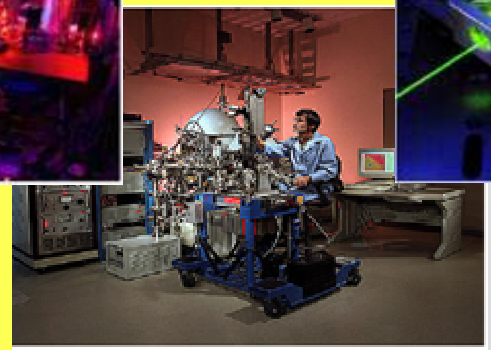
EMSL Lab Instruments

High Field Magnetic Resonance Facility

High Performance Mass Spectrometry Facility

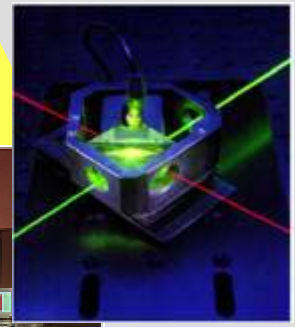
Chemistry and Physics of Complex Systems Facility



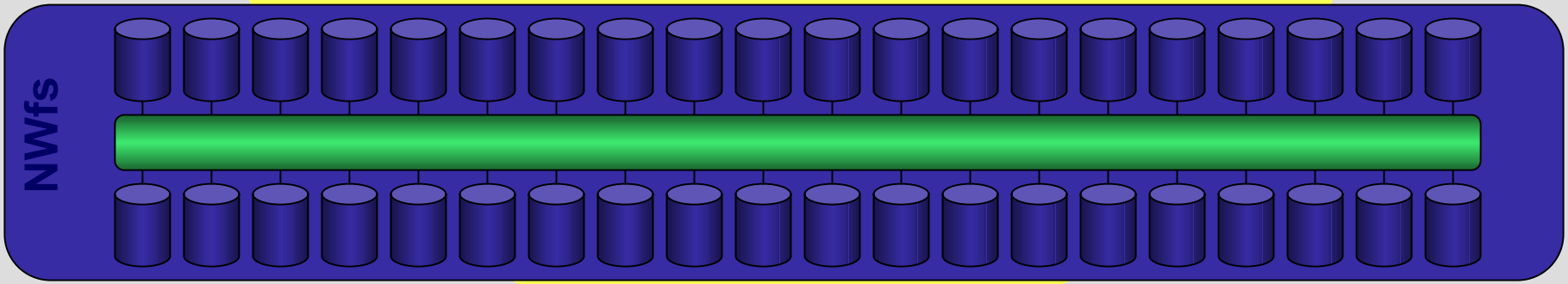


Interfacial and Nanoscale Science Facility

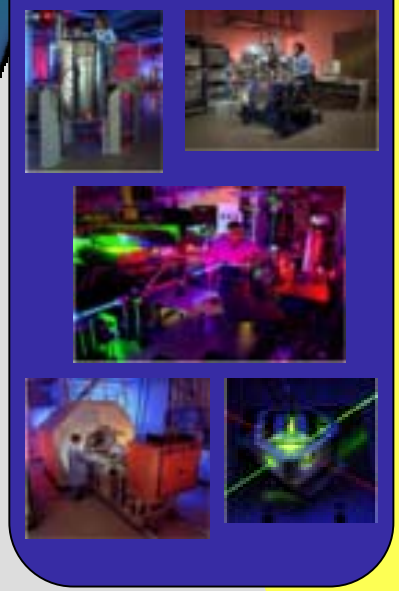
Environmental Spectroscopy and Biogeochemistry Facility



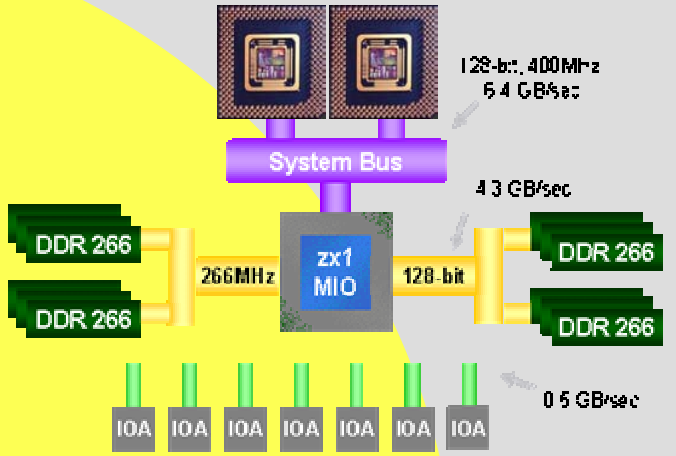
END-TO-END GIGABIT ETHERNET TCP/IP NETWORK

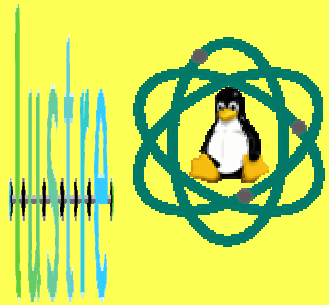


HPCS2

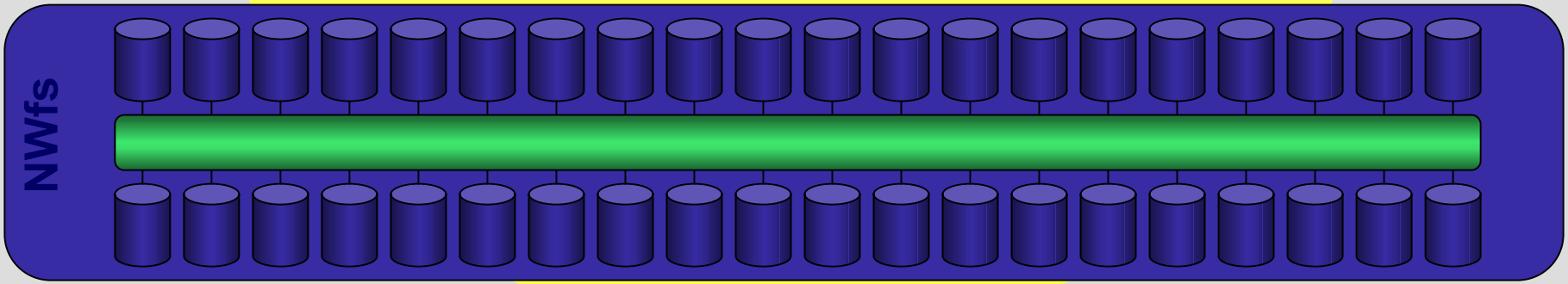


1900 Madison Processors
NWLinux
Quadric's Elan4
hp's ZX1 chip set
Gig-E connection to all nodes
53 TB of global storage
200 TG local disk
Lustre Lite File System

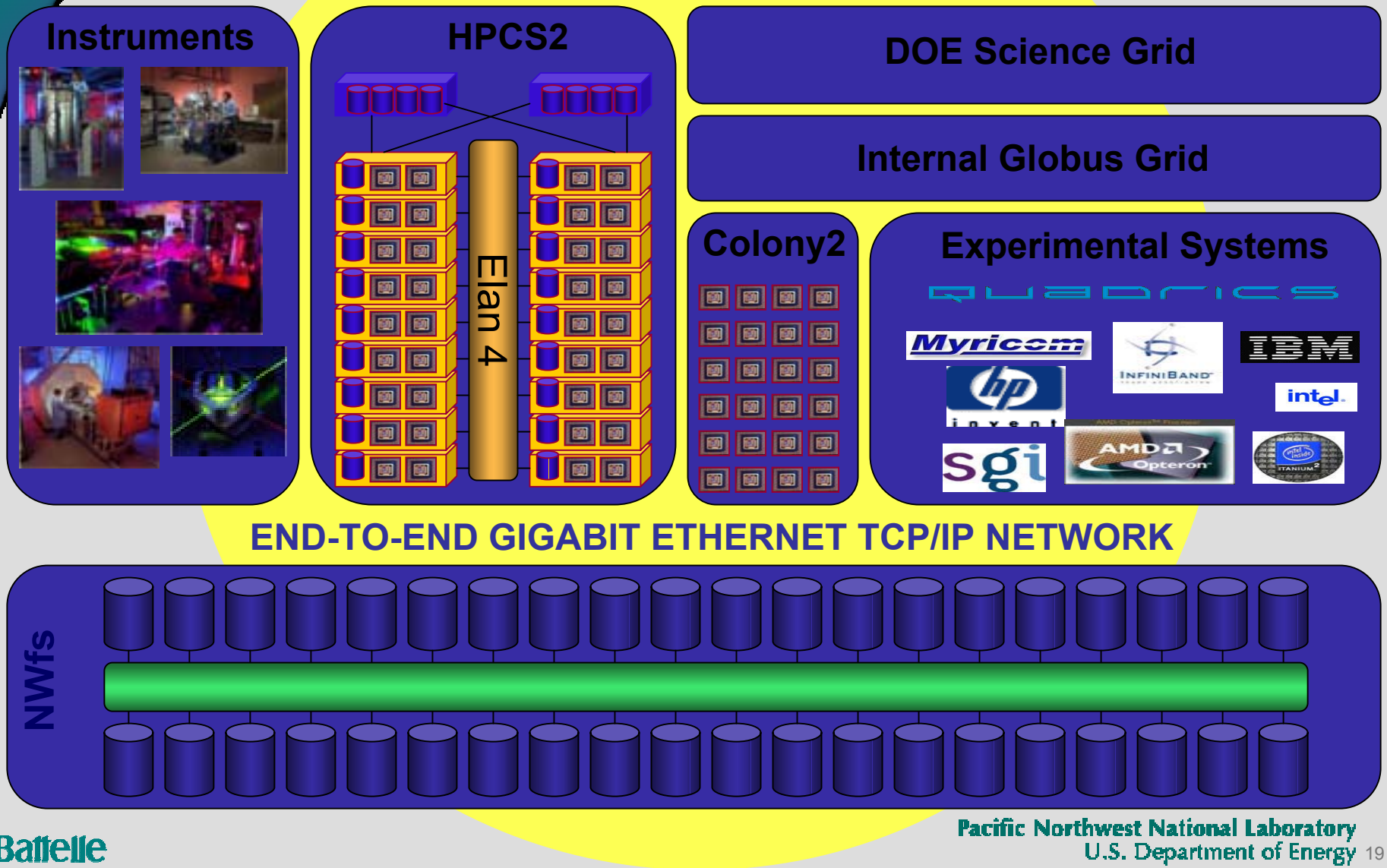




END-TO-END GIGABIT ETHERNET TCP/IP NETWORK



The Grid


Instruments
HPCS2
Elan 4
DOE Science Grid
Internal Globus Grid
Colony2
Experimental Systems
QUADRICS
Myricom
INFINIBAND
IBM
hp
intel
sgi
AMD
Opteron
ITANIUM 2
END-TO-END GIGABIT ETHERNET TCP/IP NETWORK
NWfs

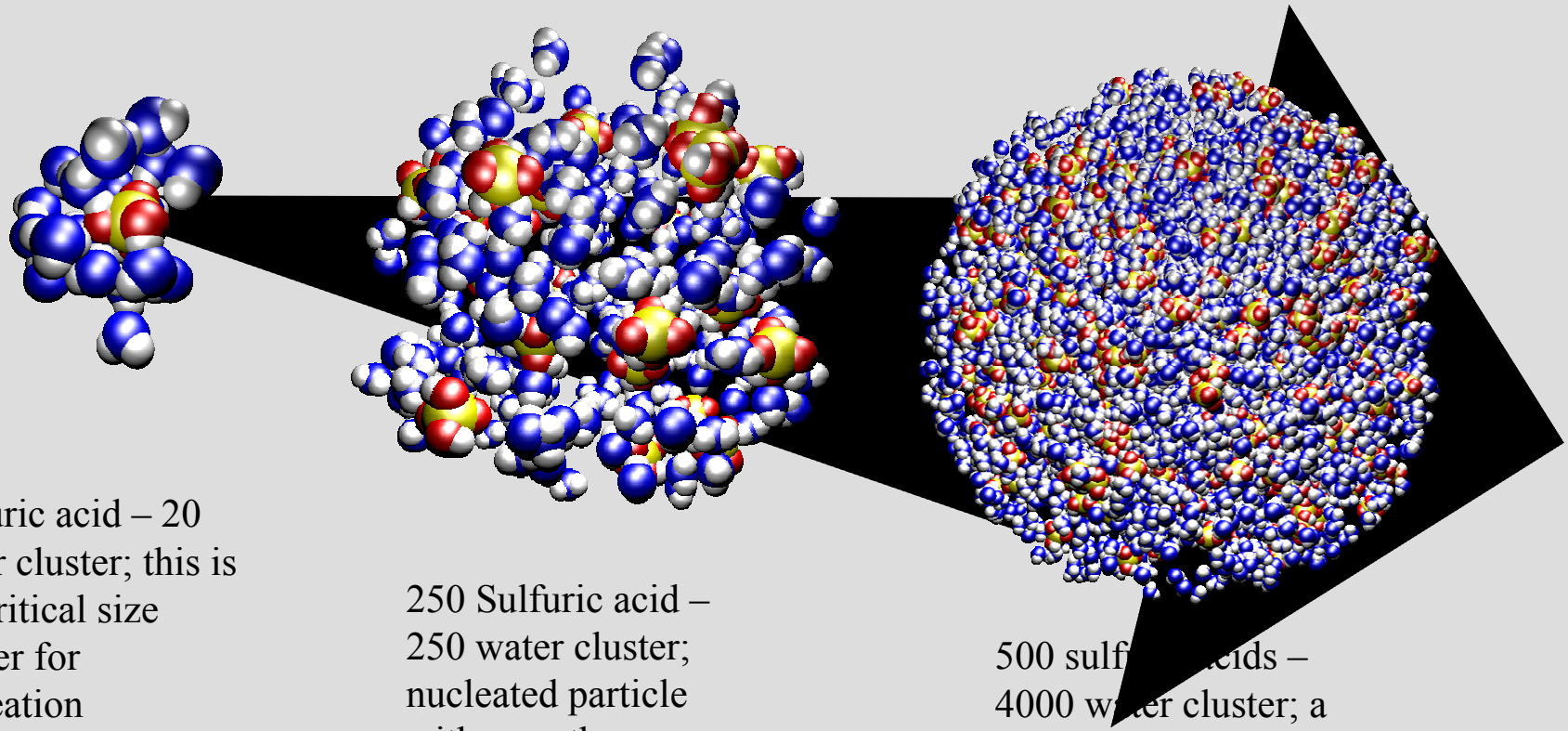
DOE/PNNL Science Drivers

- Chemistry
- Subsurface Transport
- Atmospheric Transport
- Computational Biology

HPCS1
.25TF

HPCS2
11TF

HPCS3



Sulfuric acid – 20 water cluster; this is the critical size cluster for nucleation

250 Sulfuric acid – 250 water cluster; nucleated particle with growth

500 sulf[illegible]cids – 4000 w[illegible]er cluster; a nanodroplet of sulfulric acid

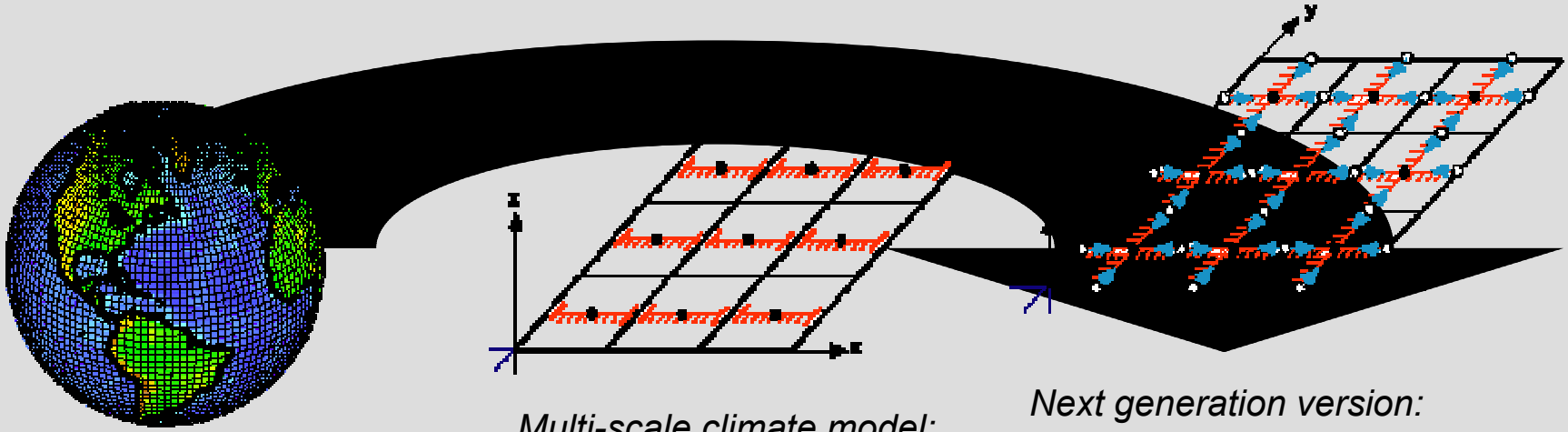
Chemical Dynamics: (Greg Shenter)

Nucleation for aerosols

HPCS1
.25TF

HPCS2
11TF

HPCS3



"Traditional model"
- Coarse grid resolution
- Cloud parameterization averaged over entire box

Multi-scale climate model:
- Grid resolution ~ 400 km
- Cloud properties computed explicitly on embedded 2D model at scale of 4 km
- Computational burden increases by factor of 200
- Can be run for 1 or 2 years of simulated time

Next generation version:
- Two orthogonal 2D models
- Central point is truly 3D
- Cloud model information is passed across outer grid boundaries at end of time step
- Computational burden will be more than double previous model

Atmospheric Chemistry and Modeling: (Tom Ackerman)

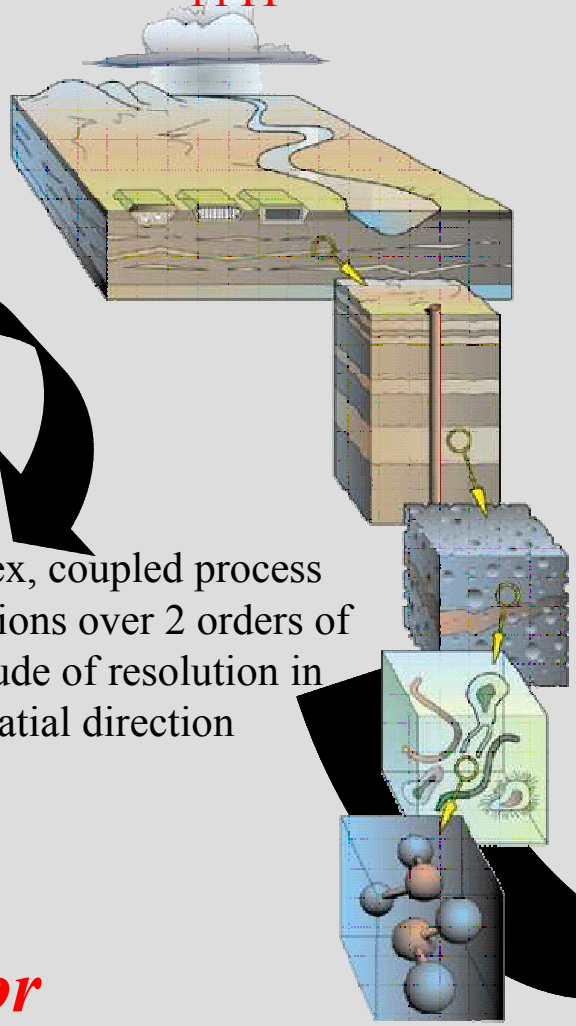
Multi-scale climate models with increased fidelity

HPCS1
.25TF

HPCS2
11TF

HPCS3

Subsurface simulations approaching 2 orders of magnitude of resolution in each spatial dimension



Complex, coupled process simulations over 2 orders of magnitude of resolution in each spatial direction

Complex, coupled process simulations approaching 3 orders of magnitude of resolution in each spatial direction

Contaminant behavior

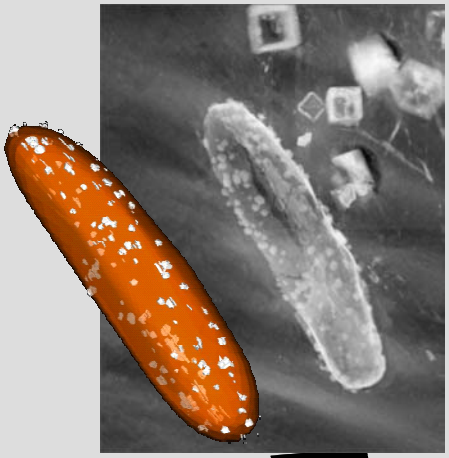
Fundamental scientific basis for field-scale predictions

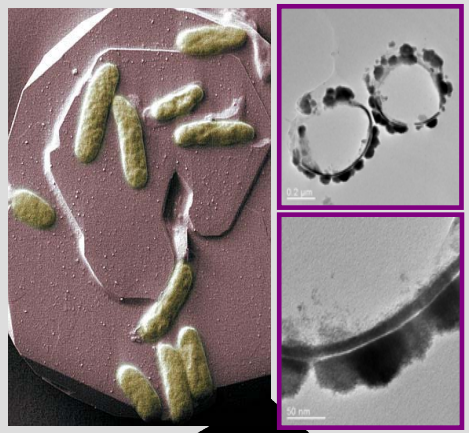
HPCS1
.25TF

HPCS2
11TF

HPCS3

We couldn't compute spatial details on MPP1, could generate basic geometry





Begin to mo patial details on an i al cell basis with "sim biochemistry.

Potential del multi-cell commu with "complex" bio ry within an onmental context.

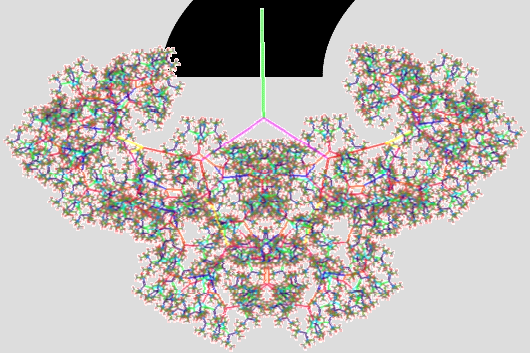
Computational Biology: (Harold Trease)

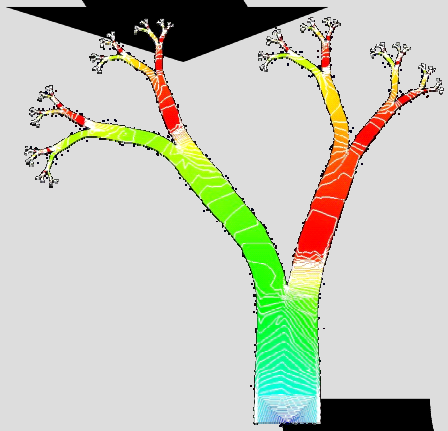
Modeling and Simulation of the Bioremediation of Heavy Metal Waste by Shewanella MR-1 using Multi-Scale Computational Methods

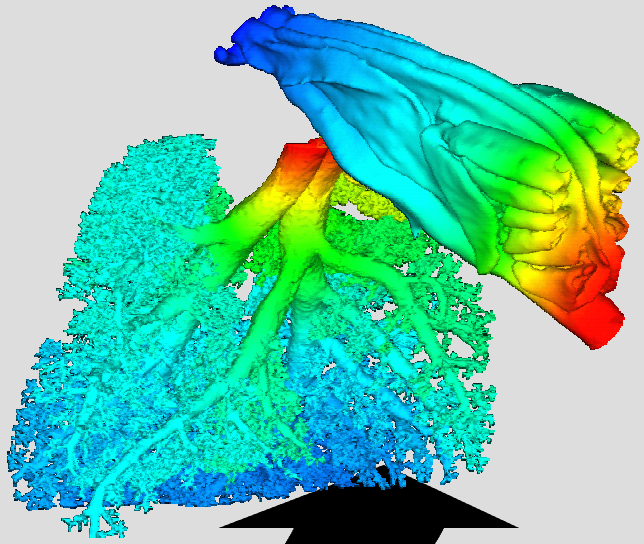
HPCS1
.25TF

HPCS2
11TF

HPCS3







Simple 2-D geometries with fluid dynamics possible.

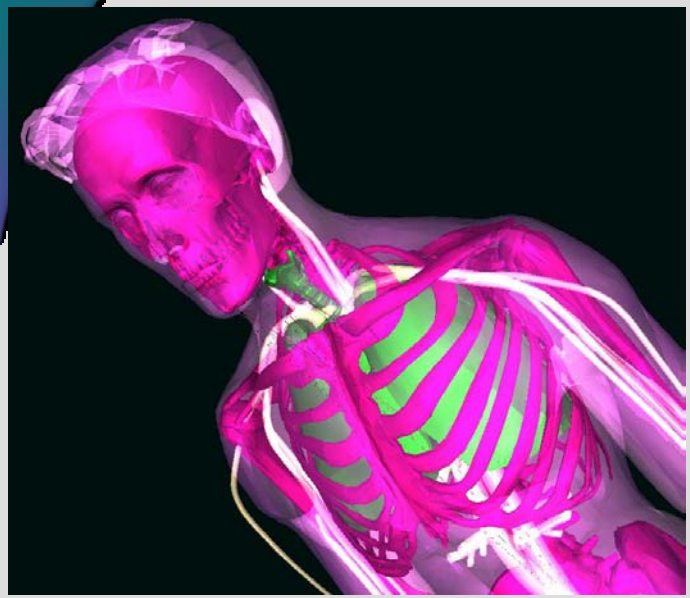
More complex 3-D geometries and fluid dynamics and transport.

Coupled u[illegible]d lower respir[illegible]tract, with: 3-D geometries, fluid dynamics, tissue mechanics, transport and cellular response

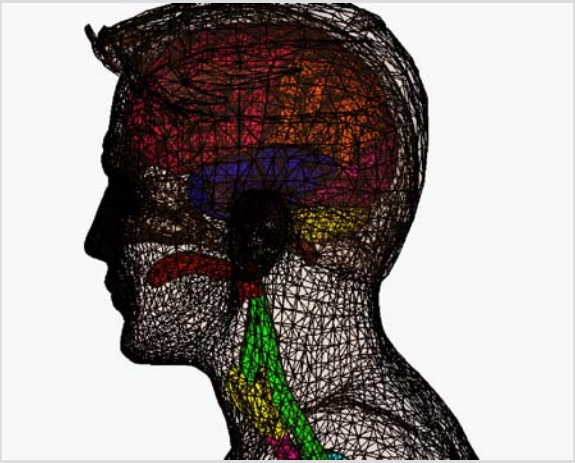
Computational Biology: (Harold Trease)

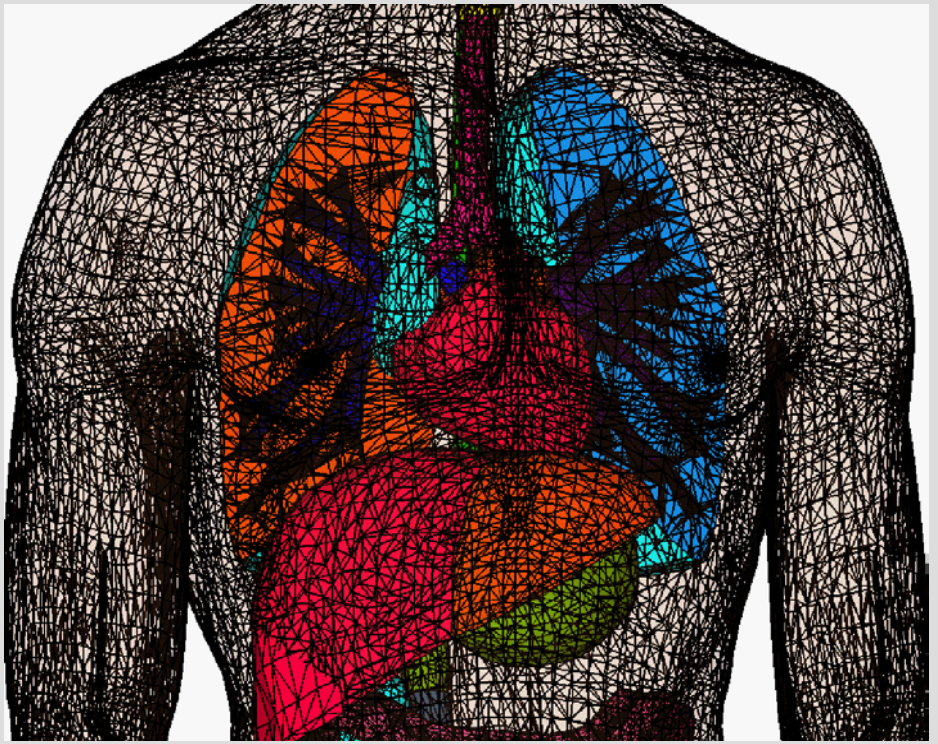
Virtual Respiratory Tract Modeling and Simulation with Particle Deposition and Chemical Vapor Transport

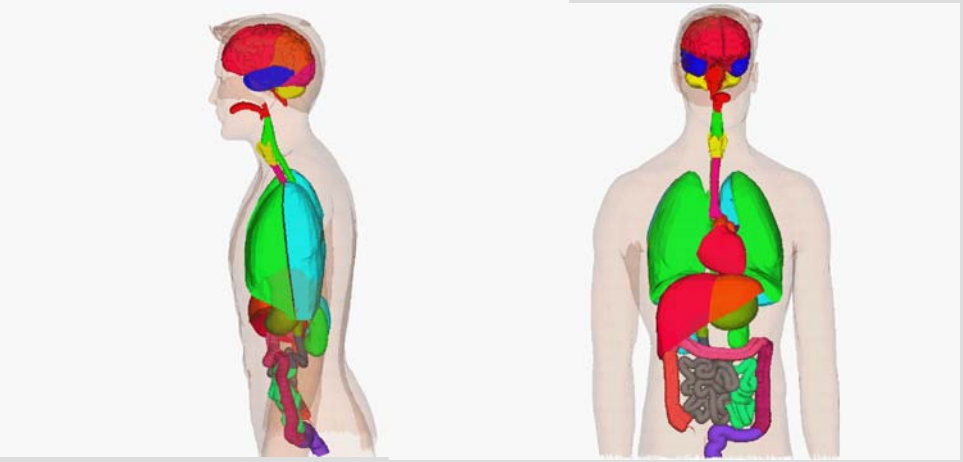
Modeling and Simulation of Complex Biological Systems



(Frankenstein meets Extreme Computing)

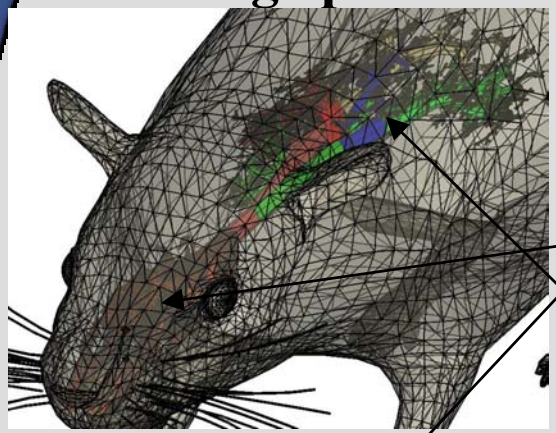






The Virtual Respiratory Tract (VRT)

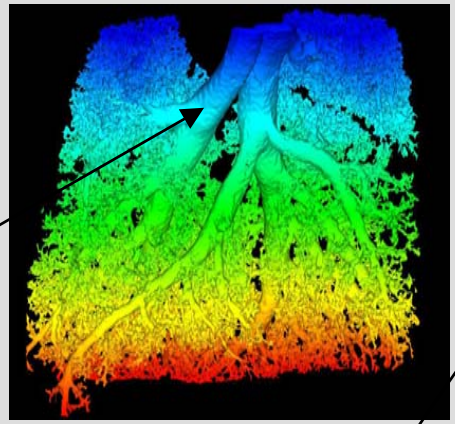
- **The team: Rick Corley, Chuck Timchalk, Kevin Minard, Harold Trease, Lynn Trease)**
  - **Multiple Species, Multiple individuals/species**
  - **Image processing, Quantitative geometry, Simulation, Validation**



Rat respiratory system:

URT

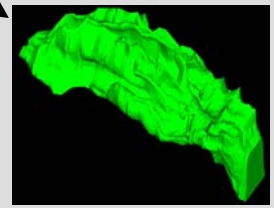
LRT



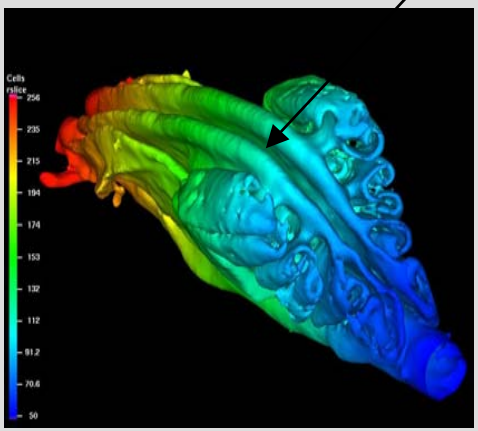
Human respiratory system:

URT

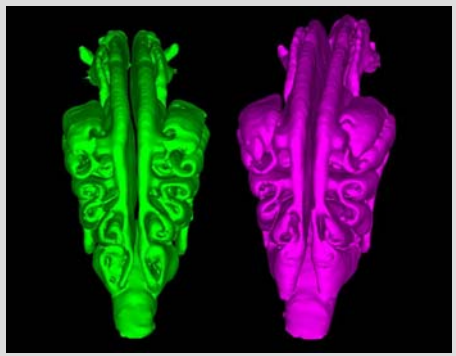
LRT

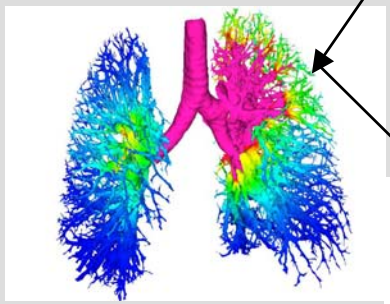


Human nose

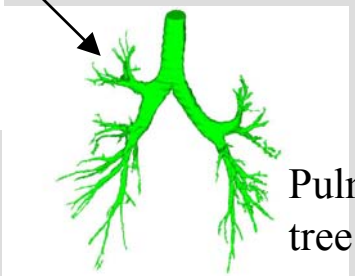


Multiple rats



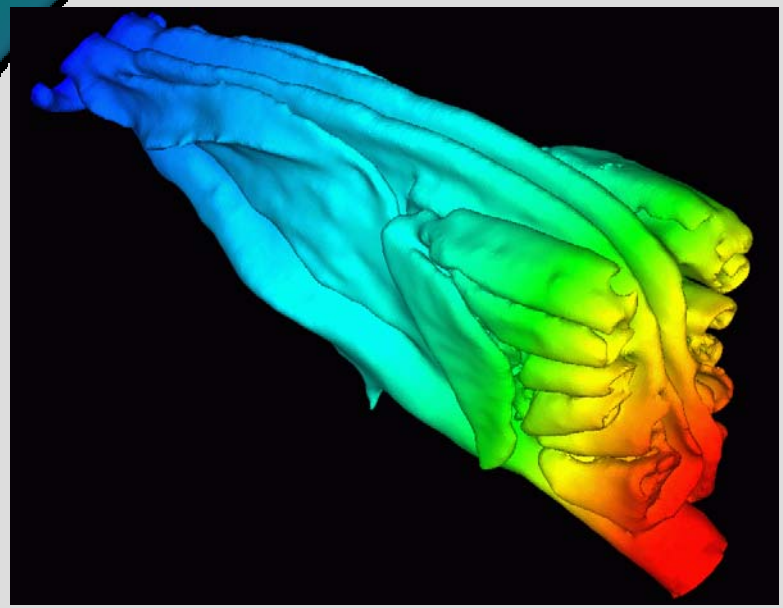


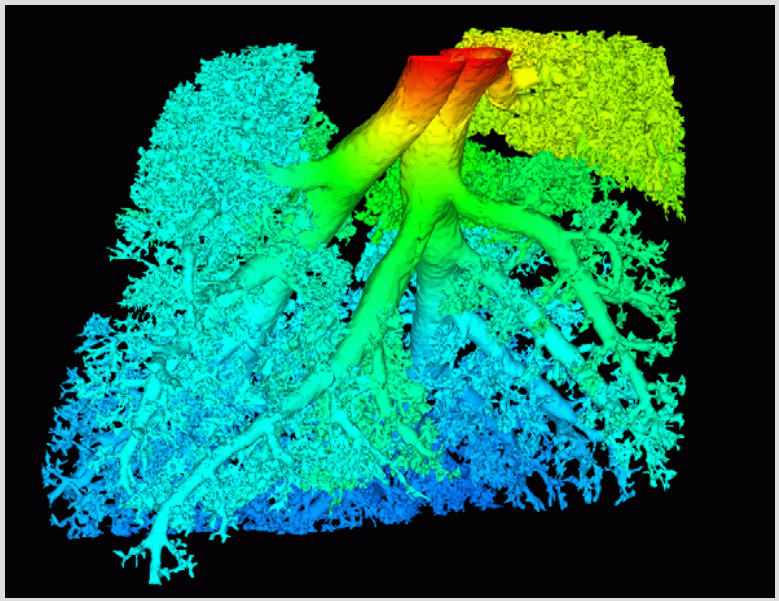
Pulmonary and arterial trees

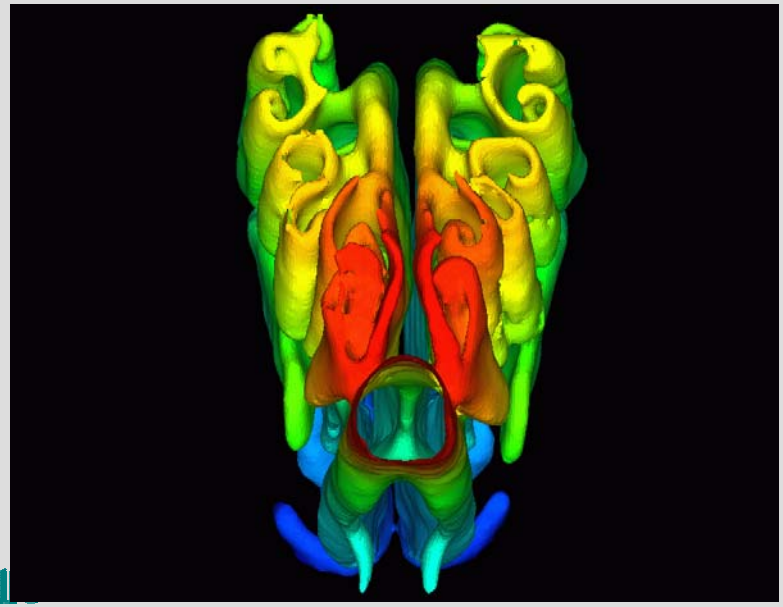


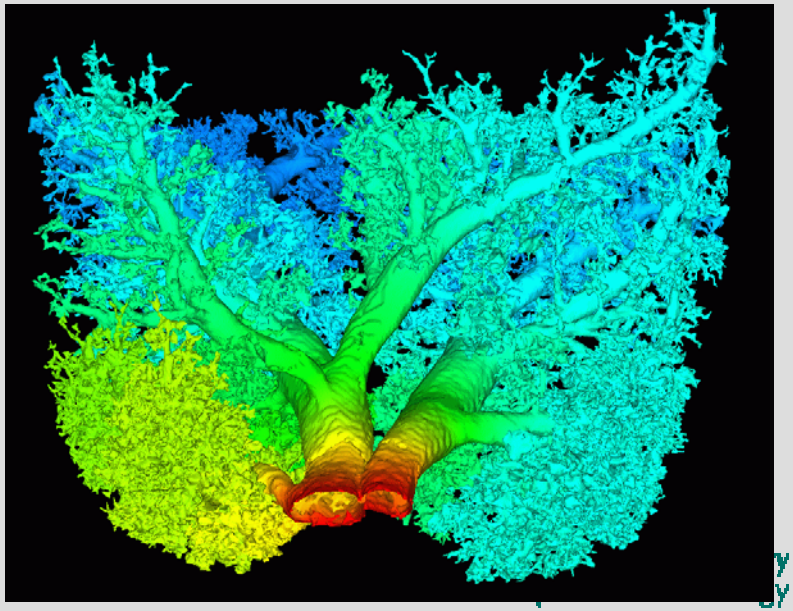
Pulmonary tree

Reconstructions of a rats nose and lungs from NMR data.









Soapbox Issues

- We need a high-level parallel language.
  - Locality of data (may be a RTS issue)
  - Communication libraries
  - Language

- Dynamic Debugging Environment:
  - Why can't we have one available before a machine reaches the end of it's lifetime ??

Soapbox Issues

- Time to solution as a metric for success. Better mathematics and numerics don't always lead to increased machine performance and higher efficiency.
- Scalability: Approximate linear scaling of mesh based methods, with number of processors and problem size, is guaranteed unless you screw it up.

Soapbox Issues

- If the goal is to use terascale machines to solve science problems, then application programmers need access to development cycles and dedicated terascale cycles.

- For sustained performance and efficiency we need a stable code development environment. [BUT: If you can't run with the big dogs you better stay on the porch.]

Lessons Learned

- Scientific Discovery Through Advanced Computing
- Time to solution as a metric for success.

- The shared memory parallel programming paradigm is the correct model for mesh based computing.
- Parallel languages (we need one)
- Debugging environments / distributed graphics/visualization

- Access to xxxxflop/xxxxbyte resources.

PNNL/DOE Funding and Acknowledgements

- SciDAC TSTT Center (Terascale Simulation Tools and Technology) [DOE OASCR/MICS]
- Microbial Cell Biology [DOE OBER/GTL] , Computational Biology Simulation Frameworks [DOE OASCR/MICS]
- Virtual Respiratory Tract Project [PNNL/LDRD, NIH]